The simplest way to start is to build a linear model, that is, to add all variables together in a linear form. For example, we can formulate a simple multi-factor alpha model as

$$\alpha = b_0 + b_1\text{PROF} + b_2\text{EQ} + b_3\text{MQ} + b_4\text{VALUE} + b_5\text{PM} + b_6\text{MS} + \epsilon, \quad (4.19)$$

where α is future stock returns, PROF is profitability, EQ is earnings quality, MQ is management quality, VALUE is value, PM is price momentum, MS is market sentiment, b_k is the impact of each theme or factor on returns, and ϵ is everything else that impacts stock prices.

We have intuitively identified themes that impact stock returns. What are the signals in each theme and how can we construct them? The following section describes each theme with its fundamental intuitions and associated signals.

4.7.2 Building Signals for Each Theme

In this subsection, we show the building process for each theme in this section, including rationales, factor definitions, and formulas.

4.7.2.1 Profitability

How do we know a stock's return will outperform the market? At end of the day, a stock is simply a share in the ownership of a business. Stock prices can fluctuate due to a multitude of factors in the short term, but over the long term, the stock price and the value of a business tend to move in the same direction. Common sense suggests that the more profitable the business, the greater its ability to create value for investors. Unsurprisingly, statistical research has proven that companies with superior profitability levels tend to produce above-average returns for investors. If a company is consistently making above-average profits, this generally indicates that it has superior business qualities, such as a differentiated brand, better technology, and/or a more innovative management team.

Given the transparency and availability of public financial filings nowadays, it is fairly easy to identify profitable companies. For illustration purposes, we present three signals for the profitability theme.

Return on Equity This signal measures the value created for shareholders, where return is measured by net income from the income statement.

$$ROE = \frac{NI}{Equity} = \frac{\sum_{t=1}^{4} NI_t}{Equity_t}.$$

The numerator is called the trailing twelve-month (TTM) value or annualized value for net income.

Cash Flow to Assets This signal measures the capability of cash flow generation from business based on total assets, where cash flow from operation is from cash flow statements.

$$CFO2TA = \frac{CFO}{TA}.$$

Earnings per Share Growth This ratio measures the time series path of earnings growth over the most recent quarter and the same quarter in the previous year.

$$EPS_g = \frac{EPS_t}{EPS_{t-1}} + \frac{EPS_t}{EPS_{t-4}}.$$

4.7.2.2 Earnings Quality

All else being equal, it is desirable to invest in companies with higher profits. However, we need to answer several questions: Are those earnings "real"? Where do they come from? Are they sustainable? The quality of earnings is very important for stock prices. Simply put, earnings quality means that the earnings are generated from a company's sound and reoccurring business operations, especially its core business, rather than creative accounting or extraordinary items. The theme of earnings quality is often used to assess the accuracy and sustainability of historical earnings as well as the achievability of future projections. Evaluating earnings quality will help investors make judgments about the certainty of current income and the prospects for the future.

Earnings refers to sales minus costs. First, good quality earnings should be real (bona fide) earnings. Unfortunately, there are incentives to engage in creative accounting. Many public companies link earnings per share of the company to the compensation of the senior management team, either directly or indirectly. In quantitative investing, the following signals are used by professional investors to measure earnings quality.

Accruals This ratio measures the difference between cash flow and net income (Sloan 1996). Cash flow and net income should generally go in the same direction. For a public company, a gap with a high net income but negative cash flows from operations, deviating far from its peers or industry norm, may signal low earnings quality.

$$Accruals = \frac{NI - CFO}{TA}.$$

Sustainable Cash Flow This ratio measures the cash flow growth path over the last 3 years. Companies with stable and positive growth rates are preferred.

$$CFO_g = \frac{\frac{CFO_t}{CFO_{t-1}} + \frac{CFO_t}{CFO_{t-4}} + \frac{CFO_t}{CFO_{t-8}}}{sd(CFO_{t,t-1....,t-12})}.$$

4.7.2.3 Value

Value signals measure the difference between market and accounting valuations of a public company. We can measure value signals from three different angles: net income, cash flow, and balance sheet. Note that the three angles will yield different information regarding how cheap the company is, although there is some overlapping: the denominator is the same, and there is a certain relationship between net income, cash flow, and book value.

Book Value to Market Cap This is the value signal from the balance sheet. It is measured by the accounting value of the company's common shares divided by the market value of those common shares (see Fama and French (1992)).

$$B/P = \frac{\text{Book Value}}{\text{Market Capitalization}} = \frac{\text{Bool value per share}}{price}.$$

Earnings to Price This is the value signal from the income statement. This reflects the earnings support per share for the stock price: the price comes from earnings and how the market evaluates per dollar of earnings.

$$E/P = \frac{NI}{\text{Market Capitalization}} = \frac{EPS}{Price}.$$

Cash Flow to Price This is the value signal from the cash flow statement. This reflects the cash flow support for the market price. Usually, cash flow is more difficult to manipulate and more transparent than earnings.

$$CFO/P = \frac{\text{CFO per share}}{Price}.$$

We see that the numerators of the above three value signals are derived from different financial statements: balance sheet, income statement, and cash flow statement. Since these three statements focus on different aspects of a public company, each value signal captures a different angle of valuation.

4.7.2.4 Management Capability

It is very challenging to evaluate companies' management teams. It is even more challenging to assign a rating score to the senior management team of a public company. Happy families are all the same: a great team is capable and honest and

works to increase shareholders' value. However, in the real world, many management teams work for their own benefits, frankly speaking. How can we quantitatively evaluate public companies' senior management teams? While it is hard to tell who is really good, one way is to try to identify how far the team deviates from their peers. We can evaluate the CEO and her team from two perspectives: (1) How did they acquire money? and (2) How did they spend the money? Both aspects require strategic leadership and capability, which have significant impacts on the overall performance of the company.

The answer to the first question—where did the CEO get the money?—can be quantified using a ratio with external financing (externalFIN). External financing occurs when a public company issues either stocks or debts to finance their projects and spending. In general, organic internal financing is regarded as healthy, while too much external financing is regarded as unhealthy because it indicates the company does not have enough internal resources to leverage. Moreover, equity issuance will dilute per share metrics, and external debt issuance could have serious consequences when things take a turn for the worse (see, e.g., Loughran and Ritter (1995) and Richarson and Sloan (2002)).

externalFIN This can be measured by two signals. One is the current ratio, external financing scaled by total assets. The other is growth over time, the change in external financing.

$$exFIN = \frac{\text{external Financing}}{TotalAsset}$$

$$exFIN_g = \frac{exFin_t}{exFin_{t-1}}.$$

Another important decision the senior management team needs to make is how to invest money in projects. While investments are necessary for profitable projects, overspending is always a bad signal for the company. Regardless, many CEOs have a tendency to build a *bigger* company instead of a *stronger* company. Some CEOs go even further to build an empire in the industry or even go beyond their expertise to cross into other industries. The latter happens often when a company has been running well and becomes overconfident, undertaking large expansions into unfamiliar areas. While we cannot know each company's projects, we do know the monetary value of capital expenditure, which can be used to quantify overspending. If the capital expenditure ratio and growth are far greater than peer companies in the industry, it is usually a case of overspending and will eventually be penalized by the market due to low or even negative returns on investment – indicating that the senior management team has made bad decisions about projects. Classic academic studies on this topic include Jensen (1966), Titman et al. (2004), and Cooper et al. (2008).

Capital Expenditure This can be measured by the following ratios: total capital expenditure scaled by total assets and change in capital expenditure.

$$CAPX2TA = \frac{\text{Capital Expenditure}}{TotalAsset}$$

$$CAPX_g = \frac{\text{Capital Expenditure}_t}{\text{Capital Expenditure}_{t-1}}.$$

4.7.2.5 Momentum

Momentum signals are based on past price movements. In quantitative investing, for strategies with moderate or longer holding periods (ranging from months to years), the industry has employed momentum with different look-back periods, such as 6 or 12 months.

$$PM6m = \frac{P_t}{P_{t-T}}, \qquad \text{where T= 6 months.}$$

Why does the momentum strategy work? There are many academic explanations (e.g., Jegadeesh 1987, Jegadeesh and Titman 1993), such as behavioral overreaction/underreaction to news, the market payoff for taking greater risk, the extra push on price from herding effects of holding winners and avoiding losers, etc. Fundamentally speaking, if a firm's stock has been doing well for the past few months, it may indicate some business value, such as market share, new products, etc. These fundamental edges will last for a while, and learning and replication by peers in the same industry may take a while.

It is intuitive that momentum signals will work better for an up trending market. For example, the same momentum signal works effectively in the US stock market but not in the Japanese stock market because the former has been up and the latter has been flat. However, it should be noted that when the market turns around, the momentum strategy can cause huge losses. Given the frequent price changes in the market, momentum is a relatively short-term and high-turnover theme.

4.7.2.6 Market Sentiment

Market sentiment refers to the "mood" of the market. It can include (1) forecasts from massive professional analysts for public companies, such as estimates for EPS, and (2) investors' overall attitude towards a financial market. In the context of quantitative investing, for the former, the industry employs analyst estimates' consensus for momentum and diffusion; for the latter, one signal employed widely by the industry is shorting activities.

Sell-side research analysts estimate the performance of public companies. They are usually professionals who follow up with companies for years. They visit the companies, meet with senior management, and make informed evaluations. The key evaluation metrics are earnings per share (EPS), cash flow per share (CFPS), the

net present value (NPV), and buy/hold/sell recommendations. There are numerous studies about analysts' estimates from both industry practitioners and academia. Some early studies include Zacks (1979), Brown et al. (1980), Abarbanell (1991) and Mikhail et al. (1999). Here we present two market sentiment signals: consensus and diffusion.

Three-Month Earnings Momentum This signal measures dynamic consensus of ups and downs of EPS revisions among analysts. The data is from IBES.

$$EM3m = EPS_t - \left[0.5\, EPS_{t-1} + 0.3\, EPS_{t-2} + 0.2\, EPS_{t-3}\right],$$

where EPS can be the average estimates for the upcoming fiscal years or quarters. Here, we treat all analysts the same and all estimates the same. Of course, there are more considerations one should make, such as lead analysts versus follower analysts, local analysts versus foreign analysts, etc. Another important aspect is accuracy versus timeliness.

Three-Month Earnings Diffusion This signal measures disagreement among analysts about the directional change of EPS for a public company.

$$EM3d = 0.7sd(EPS_t) + 0.2sd(EPS_{t-1}) + 0.1sd(EPS_{t-2}),$$

which is just a weighted average of disagreement about EPS over different periods. It has been found that more disagreement indicates more disappointing performance of the company's business, and hence downward pressure on its stock price. This is because analysts tend to have more agreement when the outlook is good, so disagreement usually indicates a bad situation.

Regarding the collective sentiment of investors towards a stock, we present short interest.

Short Interest Shares in short positions as the total number of shares floating. The data usually comes from a third-party vendor, brokerage, or custodian bank.

$$SHI = \frac{\text{short positions}}{\text{floating shares}}.$$

There are variations, such as dollar-value based or the change of SHI.

In addition to the signals mentioned above, there are other approaches measuring investor attention, such as survey-based sentiment indexes, textual sentiment from specialized online resources, internet search behavior, and non-economic factors.[5] Of course, the sentiment signal usually has high turnover as attitudes swing due to uncertainty. Sentiment signals are usually highly correlated with price movement

[5]For example, a consumer confidence index at the macro level.

signals because, for example, some analysts update their estimates based on price movements.

We have classified drivers and indicators for stock returns into themes and identified signals within each investment theme. We now need to combine those signals into a composite score as a factor or theme in a linear alpha model.

4.7.3 Signals, Themes, and Alpha: Data Treatment and OLS

In this subsection, we present an industry approach to return forecasting in the context of a stock selection strategy. Note that there may be large variations depending on investment strategy, universe, portfolio characteristics, etc.

We assume data availability and use constructed signals and themes as specified in the previous subsection. We present below a typical industry procedure for alpha building from a multi-factor analytical framework, starting from signals, then themes, and finally alpha.

1. Select factors for each theme.
2. Decide an investment horizon.
3. Use a proper industry classification.
4. Clean and treat raw signals.
   (a) missing data
   (b) errors
   (c) outliers
   (d) commensurability (compare signals apples-to-apples)
      – same range
      – demeaned by industry to remove industry bias
      – distribution
5. Before multi-factor analysis
   (a) univariate analysis: efficacy of each signal
   (b) bivariate analysis: correlation
6. OLS regression: multi-factor model

   Stage 1: form themes

   (a) coefficients, t-value, and R^2 value
   (b) weights for signals

   Stage 2: form alpha

   (a) match with features of investment strategy
   (b) critical for portfolio performance

We illustrate the above process using a stock selection strategy in the large cap segments of the US stock market. In the active institutional investment space, the large-cap part of the US stock market is usually represented by the Russell 1000

index, a conventional investment universe as well as a popular benchmark. There are several reasons that most institutional investors use the Russell 1000 rather than the S&P 500: First, the Russell 1000 has 500 more large companies and is therefore more representative of the large cap universe without sacrificing liquidity. This provides greater breadth for quantitative investing, which relies on the law of large numbers. Second, the Russell 1000's constituents are all US companies, whereas the S&P 500 includes foreign companies. Finally, the Russell 1000 has more transparent rules on index inclusion and rebalancing so that exiting and new entries are more predictable.

Universe and Signals We use Russell 1000 monthly data from January 31, 1995 to December 31, 2004. The data is available on the month-end trading date. Following the industry convention, we use *Cusip* as the id for the US companies. Any record will be identified by a cusip and date. We select the following factors for the six themes:[6]

VALUE:	S/P, B/P, E/P, CFO/EV
PM:	PM1m, PM6m, PM9m
PROF:	ROE, EBITDA/EV
EQ:	accrualsCF, CFOxInt/debt
MQ:	CAPXg, exFINg
MS:	EM9m, ED9m, EM12m, ED12m

Investment Horizon and Industry Classification Assuming the portfolio has a medium- to long-term investment horizon, we set the investment horizon to be nine-months. We use 9-month cumulative returns as the dependent variable for the multi-factor model. Regarding industry classification, we use GICS, which is widely used by institutional investors in the USA. It had ten sectors during the 10-year period from 1995 to 2004. Industry classification is very important because it defines industry characteristics, determines peer companies, and thus impacts stock selections within each industry. Accordingly, this should be reflected in signal treatment and alpha values. Industry classification is also important for portfolio construction because constraints are imposed at the industry level for risk management purposes.

For a large company in the Russell 1000, its industry exposure can be multiple and dynamic. First, a large company may have multiple lines of business across different industries. Second, when a company changes its major business, the industry classification will change. These should all be considered in investing.

[6]We use both names of S2P and S/P for the same ratio.

Raw Signals Treatment Following the procedure specified above, we clean the raw signals with error values, treat the raw signals with outliers, demean the signal values across industries, and standardize the final values to produce the same range.

- Cleaning raw signals
  - Missing values: if >30%, invalidate the signal; otherwise, fill in zeroes
  - Error values: remove and treat as missing
  - Outliers: truncation at 5 sigma and winsorization at 3 sigma
- Demean by industry
- Z-score with range −3 to 3
- Maintain the original distribution

We illustrate above using signals in the value theme. First, we show the raw signal values with the summary statistics generated by R scripts below. We see that the raw data are full of errors and outliers. If we use these raw values for investment the results will be totally meaningless. We need to process the raw signals based on the rules specified above. The values after the treatment are also displayed by R scripts below.[7]

Raw and treated signals

```
> # summary of raw value signals: S2P, B2P, E2P, CFO2EV
> dim(r1k.rawdata)
[1] 117777    280

> summary(r1k.rawdata$S2P)
    Min.  1st Qu.   Median     Mean  3rd Qu.      Max.  NA's
  0.0000   0.3202   0.6204   1.0439   1.1412  839.7784  1559

> summary(r1k.rawdata$B2P)
     Min.   1st Qu.   Median     Mean   3rd Qu.      Max.  NA's
-132.8115    0.2119   0.3638   0.4486    0.5617  113.2039  2628

> summary(r1k.rawdata$E2P)
     Min.   1st Qu.  Median   Mean   3rd Qu.      Max.   NA's
-50000.00      2.41    4.67  82.46      6.88  50000.00    751

> summary(r1k.rawdata$CFO2EV)
     Min.  1st Qu.   Median    Mean   3rd Qu.      Max.   NA's
-50000.00     4.43     7.50  230.92     11.21  50000.00  20448
```

[7]We discuss R treatment functions for signals and multi-factor model estimation in the next section.

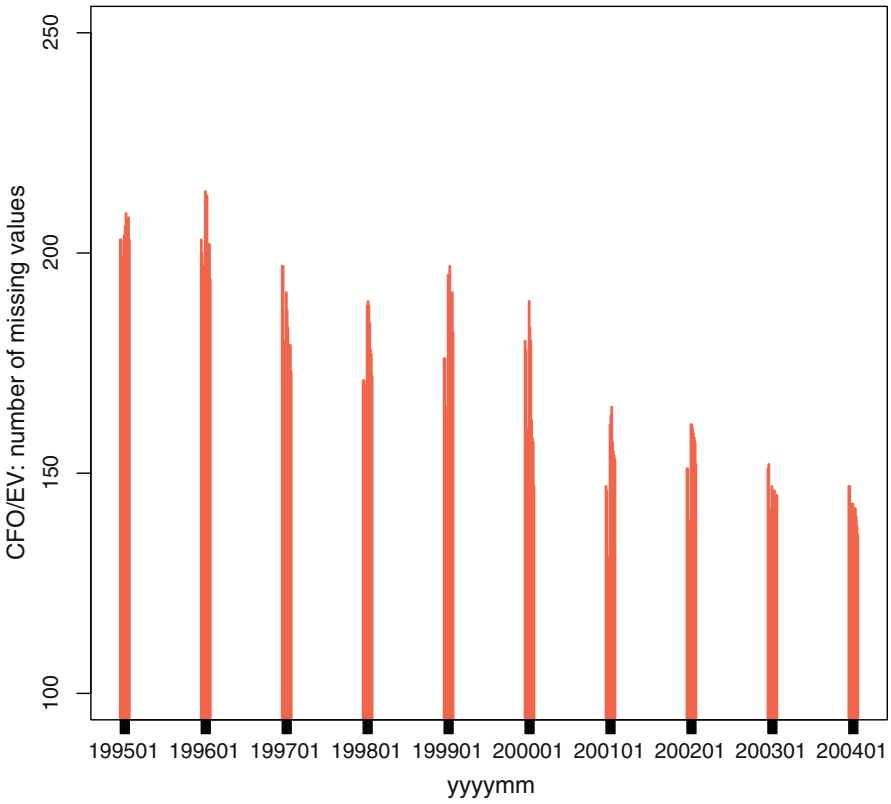

Fig. 4.5 The number of missing values for CFO/EV in the Russell 1000 universe decreased from around 200 on January 31, 1995 to around 150 on December 31, 2004

Note that pricing information is the most available and accurate, and there should be no missing values or errors, but there do exist outliers. However, accounting items are never short on such issues. Among the three financial statements, cash flow items generally have the most missing values because many items are not required to be reported. Using the value signals as an example, the CFO/EV signal has 17% missing values, while other value signals have only about 1–2% missing values. However, the good news is that the number of missing values for CFO/EV decreased dramatically from about 200 on January 31, 1995 to about 150 on December 31, 2004 (Fig. 4.5).

	S/P	B/P	E/P	CFO/EV
Missing Percentage:	*1%*	*2%*	*1%*	*17%.*

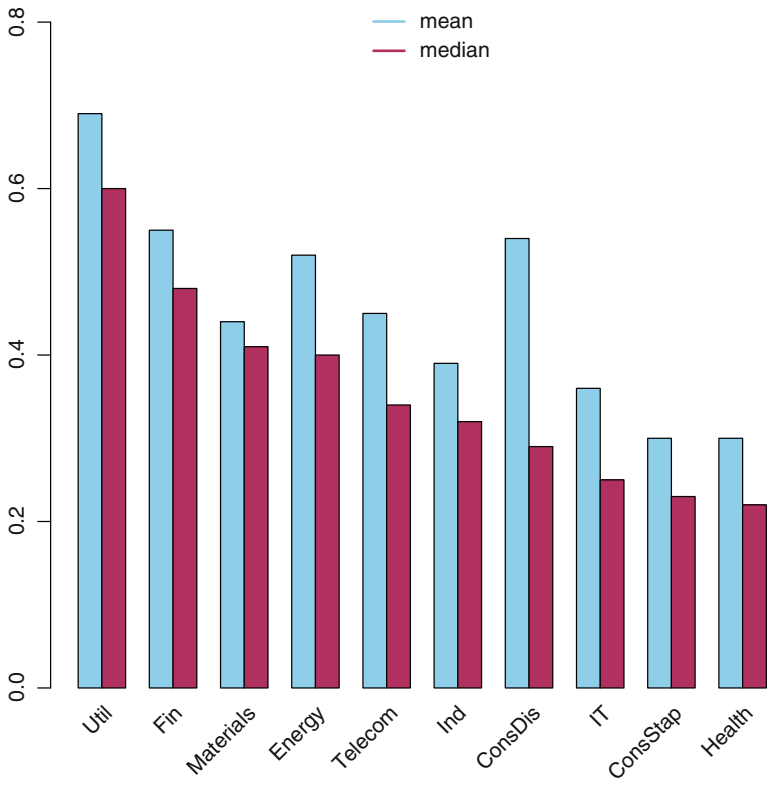

Fig. 4.6 The mean and median B/P values across GICS sectors (Russell 1000 universe) on December 31, 2004

Industries have special impacts on signal values. For example, cash flow items do not make much sense for the banking industry, and value factors are usually high for energy, banking, and utilities. This implies that without demeaning to remove industry effects, stock selection will be highly concentrated and skewed in some industries resulting in more of an industry selection than a stock selection. This is evidenced in Fig. 4.6, where plots display the mean and median B/P signal across ten GICS sectors during the period of 1995–2014. We see that, indeed, the values of B/P are very different across industries: the utilities, financials, and energy sectors have the highest median values of 0.40–0.60, while the IT, consumer staples, and health care sectors have the lowest median values of 0.20–0.25.

After dealing with the errors, missing values, and outliers in the data, and removing industry effects, we get the final treated values for each signal, with mean and median equal to zero, standard deviation equal to 1, and range equal to about -3 to 3. The box plots in Fig. 4.7 describe the distribution of final treated values for S/P, B/P, E/P, and CFO/EV on December 31, 2004 (right plot). For comparison purposes, we also have the box plots of the raw values for the four value signals (left

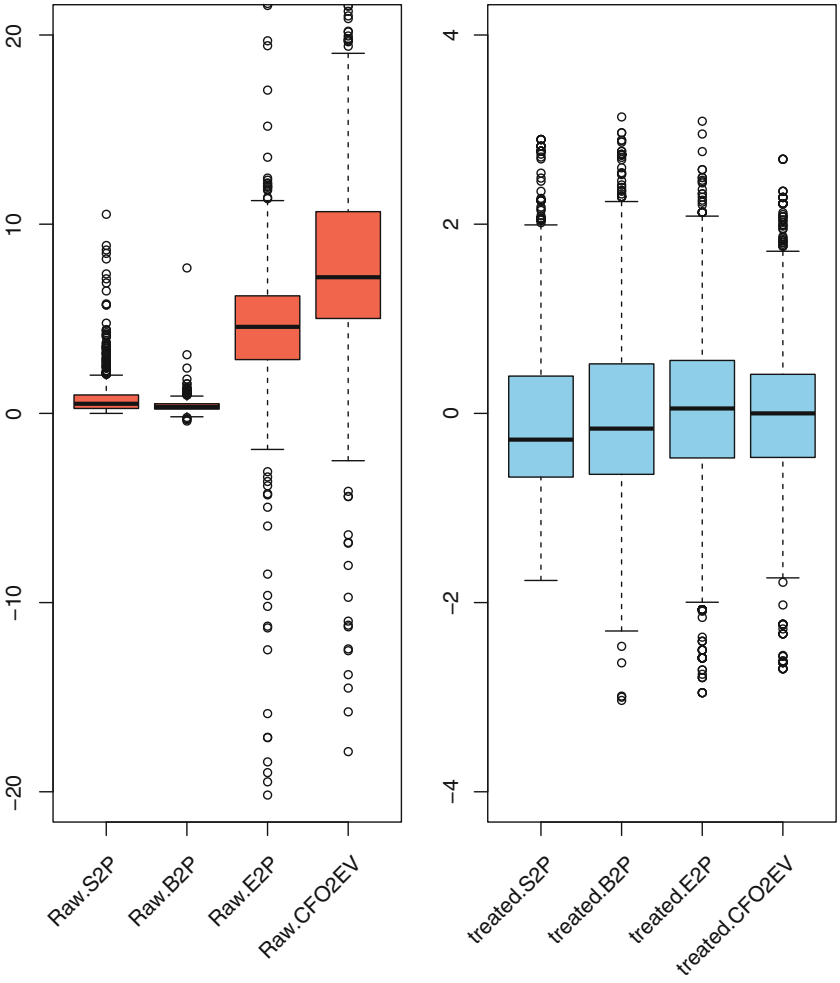

Fig. 4.7 Boxplots of value signals, raw values (left) and treated values (right), on December 31, 2004

plot). We see again that the raw scores are quite wild, the treated scores prepare us to compare signal values apples-to-apples for a stock selection strategy.

To see the factor values change after each step of the treatment, we present the correlations of values of B/P (B2P) in Fig. 4.8, where the lower triangle is for the rank correlation and the upper triangle is for the Pearson correlation. We see that the correlation is above 90%, indicating that, indeed, the signal treatment procedure removes or decreases some noise in the data while retaining the information contained in the factor.

	Raw.B2P	Truncate.B2P	Standardize.B2P	Windsorize.B2P	Neutralize.B2P	Standardize2.B2P	treated.B2P
Raw.B2P		1	1	0.98	0.93	0.93	0.93
Truncate.B2P	1		1	0.98	0.93	0.93	0.93
Standardize.B2P	1	1		0.98	0.93	0.93	0.93
Windsorize.B2P	1	1	1		0.94	0.94	0.94
Neutralize.B2P	0.92	0.92	0.92	0.92		1	1
Standardize2.B2P	0.92	0.92	0.92	0.92	1		1
treated.B2P	0.92	0.92	0.92	0.92	1	1	

Fig. 4.8 The Pearson and rank correlations between B/P values at each step of the treatment for stocks in the Russell 1000 universe on December 31, 2004

After building each signal with proper treatment of raw values, we now move on to the construction of a multi-factor model, which can be divided into two stages: stage 1—combine signals into a theme and stage 2—combine themes into an alpha.

Multi-Factor Model, Stage 1: Combine Signals into a Theme An immediate question is why we cannot put all signals in a multi-factor model rather than conduct a two-stage process. Well, the reason is simple. Because signals within a theme are highly correlated, combining signals into themes reduces the multicollinearity issue and produces more freedom in estimation. The conceptual meaning also makes sense: perhaps it is cleaner to have the factors at the theme level as the later reflect conceptual alpha sources.

Before running a multi-factor model, we investigate the efficacy of each signal with forward stock returns of 1, 3, 6, and 9 months. We continue using the value signals for illustration purposes, calculate correlations, and present them in Table 4.3. We see that overall, CFO/EV has the highest correlations, followed by E/P and S/P, while B/P has the lowest correlation. In terms of return forecasting power, in general, a correlation between signal values and forward returns around 0.01–0.05 is considered effective, 0.05–0.08 is considered very effective, and above 0.10 is suspicious and may be caused by errors, outliers, or a mistaken contemporaneous relationship. Based on the efficacy of CFO/EV, we see that cash flow is indeed the king in the US stock market, but this may not be true in other financial markets, such as in Japan, as we will see in Chap. 9. For value signals, one interesting feature is that they all have long-term effects, that is, the longer the investment horizon, the higher the forecasting power as measured by correlation. This implies that value signals are suitable for medium- to long-term investment strategies.

At this stage, we need to be extremely careful about the collinearity issue. If the correlation between signals is too high, we can just assign weights based on the results from univariate and bivariate analysis. Otherwise, we can apply OLS to the multi-factor model, use the t-value to make a judgment about the joint efficacy of factors, and then decide the weights based on univariate, bivariate, and multi-factor analysis. We give an example below for value factors. The correlations between value signals of S/P, B/P, E/P, and CFO/EV are listed in Table 4.4, where the upper triangle is the Pearson correlation and the lower triangle is the rank correlation. We see that for B/P and S/P, the Pearson and rank correlations are 0.46 and 0.50, respectively,

Table 4.3 Pearson and rank correlations of value signals with forward returns for stocks in the Russell 1000, based on monthly data from January 31, 1995 to December 31, 2004

	Retf1	Retf3	Retf6	Retf9		Retf1	Retf3	Retf6	Retf9
Pearson cor.					**Rank cor.**				
B/P	0.01	0.01	0.01	0.01	B/P	0.01	0.01	0.01	0.01
E/P	0.03	0.04	0.04	0.04	E/P	0.03	0.04	0.05	0.05
S/P	0.02	0.02	0.02	0.03	S/P	0.01	0.02	0.03	0.03
CFO/EV	0.03	0.04	0.05	0.05	CFO/EV	0.02	0.04	0.05	0.06

Retf1 is one-month forward stock returns

Table 4.4 Pearson and rank correlations between value signals for stocks in the Russell 1000, based on monthly data from January 31, 1995 to December 31, 2004

	B/P	E/P	S/P	CFO/EV
B/P	1	0.18	0.46	0.28
E/P	0.24	1	0.28	0.36
S/P	0.50	0.34	1	0.36
CFO/EV	0.34	0.38	0.42	1

Table 4.5 OLS coefficients (t-values) of value signals in the Russell 1000, based on monthly data from January 31, 1995 to December 31, 2004

	Retf9	y=Retf9	y=Retf9	y=Retf9	y=Retf9
B2P	0.0026 (2.10)				−0.0064 (−4.50)
E2P		0.0180 (14.32)			0.0122 (8.92)
S2P			0.0115 (9.19)		0.0054 (3.67)
CFO2EV				0.40219 (15.90)	0.0170 (10.88)

The first four columns are for each single variable and the last column corresponds to the multi-factor model (4.20)

which are high enough to cause collinearity in a multi-factor regression. Now, a natural question is, how high of a correlation is serious enough to cause collinearity? While the answer varies case-by-case, the rule of thumb is about 0.40 for OLS.

Now we run a multi-factor regression using different investment horizons.

$$R_F = b_0 + b_1 S2P + b_2 B2P + b_3 E2P + b_4 CFO2EV + \epsilon, \tag{4.20}$$

where R_F is forward returns at the stock level. Here we skip the time t as we apply OLS to the entire data set from January 31, 1995 to December 31, 2004. We present the estimates for coefficients in Table 4.5 with the t-values in parentheses. For comparison purposes, we have OLS estimates for each value signal in a single-factor model and then all value signals in a multi-factor model.

Connecting with theoretical explorations of OLS for multi-factor models in previous sections, we now have an industry model and real-world data to discuss in detail the OLS estimates for a multi-factor model for stocks in the Russell 1000 universe. We focus our discussion on coefficients, t-values, R^2, and BLUE

conditions. The R scripts below produce the OLS regression results for the 4-factor model (4.20).

OLS, value signals

```
> #Summary of OLS of the 4 value signals:
>  summary(lm(formula = retf9 ~ S2P + B2P +E2P+CFO2EV,
                     data = r1k.signals))
Residuals:
    Min      1Q  Median      3Q     Max
-1.1468 -0.2085 -0.0114  0.1761 27.6524

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.097010   0.001242  78.135  < 2e-16 ***
S2P          0.005433   0.001482   3.666 0.000246 ***
B2P         -0.006449   0.001432  -4.503  6.7e-06 ***
E2P          0.012175   0.001364   8.923  < 2e-16 ***
CFO2EV       0.017002   0.001562  10.881  < 2e-16 ***
---
Residual standard error: 0.4094 on 108725 degrees of freedom
  (8162 observations deleted due to missingness)
Multiple R-squared:  0.003336,Adjusted R-squared: 0.003299
F-statistic: 90.97 on 4 and 108725 DF,  p-value: < 2.2e-16
```

Note that since we apply the same treatment process for all signals, the standard errors for coefficients are very similar. Because of collinearity issues, particularly between B/P and S/P, the coefficient for B/P is negative, so the effects are rendered by S/P because it has greater forecasting power as we observed in the correlation table. This also implies that univariate and bivariate analysis are very important before we run a multi-factor analysis. Note also that the coefficients are all significant, and the order of effects from the OLS results is the same as measured by correlation: CFO/EV has the highest coefficient, followed by E/P and S/P. The R-squared values are generally very low for a multi-factor model of a stock selection strategy, usually falling below 5% and often below 3%. Here we see that value signals together can only explain 0.3% of the nine-month forward returns, even though all signals are very significant based on t-values.

This also implies that over 99% of the information is contained in ϵ, the so-called noise in (4.20). Since the left-hand variable, the response variable–forward returns, is calculated based on prices, the error term must also relate to prices. The value signals all have denominators that are prices, so inevitably, the error term ϵ and signals x_k are correlated, therefore, the arise of the endogeneity issue. Based

on what we learned from previous sections, in the presence of endogeneity, OLS estimates will no longer be unbiased or consistent.

Now, we discuss the homogeneity condition for efficiency. We obtain estimates for ϵ in (4.20). For the 10-year study period, there are 393 companies that appear in the Russell 1000 index each month. Of those 393 companies, we randomly select 100 companies and calculate standard deviation of residuals for each company and correlation between different companies. The results are presented in Fig. 4.9, where the top plot is for standard deviation and the bottom plot is for correlation. We see from the top plot that while many stocks have standard deviations of about 0.20, there do exist many stocks with standard deviations that differ significantly from 0.20, indicating the violation of error terms being identical. The violation of independence is more serious: the correlations in the bottom plot range from 10% to 40%, far from being zero! Clearly, the iid assumptions do not hold in this study. In fact, the iid condition barely holds for any financial market because securities are different and are related with each other. For example, Boeing and Bank of America are different companies, and their ϵ values will be very different, while American Airlines and United Airlines are peer companies belonging to the same industry, and their ϵ values will be highly correlated.

$$\text{violation of being identical:} \quad \sigma(\epsilon_i) \neq \sigma(\epsilon_j)$$
$$\text{violation of independence:} \quad cor(\epsilon_i, \epsilon_j) \neq 0.$$

Apparently, the OLS estimator is far from being BLUE given the violations of exogeneity and errors being non-iid. However, we can still rely on OLS to estimate joint effects of signals and use this information with other analyses to make investment decisions.

We apply the same procedure for signals within each of the other themes, and summarize the results below. Note that this is only for illustration purposes. We derive all weights based on in-sample data and ignore many other aspects, such as turnover and risk characteristics.

$$\begin{aligned}
VALUE &= 0.50CFO/EV + 0.30E2P + 0.15S2P + 0.05B2P \\
PM &= -0.20PM1m + 0.30PM6m + 0.50PM9m \\
PROF &= 0.20ROE + 0.80EBITDA/EV \\
EQ &= -0.55accrualsCF + 0.45CFOxInt/debt \\
MQ &= -0.20exFINg - 0.80CAPXg \\
MS. &= 0.20ER9m + 0.20ER12m + 0.30ED9m + 0.30ED12m.
\end{aligned}$$

Multi-Factor Model, Stage 2: Combine Themes into an Alpha Now that we have themes, we focus on alpha construction. Before running the multi-factor model, we first conduct univariate and bivariate analysis.

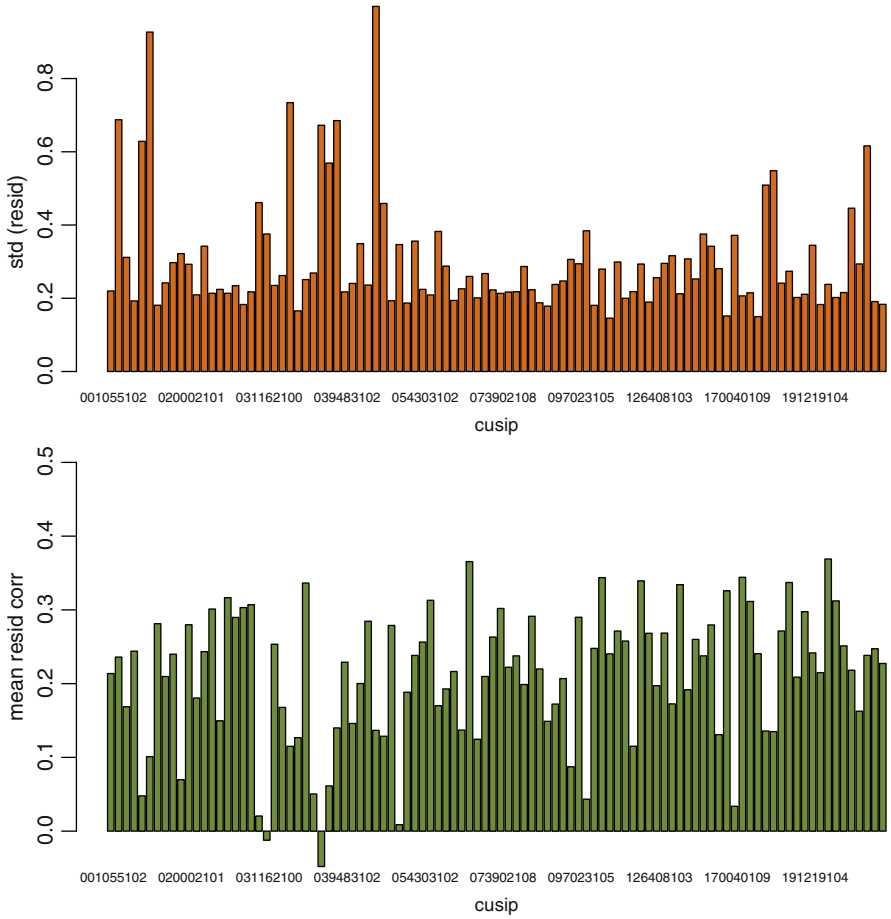

Fig. 4.9 Standard deviation of residuals (top plot) and correlation of residuals (bottom plot) of (4.20) for 100 randomly selected companies in the Russell 1000

The R scripts below yield summary statistics for each theme over the entire 10-year study period.

Univariate, summary statistics

```
##Univariate analysis, Summary of themes:
> summary(r1k.themes[,c("VALUE","PM","PROF","EQ","MQ","MS")])

    VALUE                PM                 PROF                EQ                 MQ
Min.   :-2.705059  Min.   :-3.577630  Min.   :-2.650443  Min.   :-3.1513605  Min.   :-2.447718
1st Qu.:-0.416829  1st Qu.:-0.450886  1st Qu.:-0.310891  1st Qu.:-0.3518615  1st Qu.:-0.330641
Median :-0.042301  Median :-0.016697  Median : 0.014486  Median :-0.0122225  Median : 0.060870
Mean   : 0.002291  Mean   : 0.002688  Mean   : 0.003099  Mean   :-0.0000649  Mean   : 0.001949
3rd Qu.: 0.351060  3rd Qu.: 0.439821  3rd Qu.: 0.363133  3rd Qu.: 0.3719101  3rd Qu.: 0.377287
```

```
Max.   : 3.043153  Max.   : 3.363608  Max.   : 2.738661  Max.   : 2.8282600  Max.   : 2.699267

      MS
Min.   :-3.466744
1st Qu.:-0.550293
Median : 0.000000
Mean   : 0.000887
3rd Qu.: 0.577463
Max.   : 3.505816
```

We apply necessary treatments to themes such as re-standardization, and then run correlations between each theme and a set of forward returns (Table 4.6) and between themes (Table 4.7).

As expected, the correlation scores with forward returns at the theme level are in general higher than at the signal level. Table 4.6 also shows that the rank correlations are of a similar magnitude to the Pearson correlations, indicating there are no effects from outliers. As measured by Pearson correlation, VALUE, PM, PROF, and MQ have correlation scores of 3–6%, while EQ and MS are a bit weaker, with correlation scores of 1–3%. MQ and EQ are more suited to long-term investment strategies, while MS and PM are more appropriate for short- to mid-term investment strategies. VALUE and PROF are in between.

We see in Table 4.7 that the Pearson and rank correlation scores are 74% and 68%, respectively, between VALUE and PROF; and 48% and 46%, respectively, between PM and MS. These are high enough to cause collinearity issues in a multi-factor model.

To overcome the multicollinearity issue, we employ a residual approach. We run OLS regressions of PROF on VALUE, then use residuals as a "cleaned" factor since they will still contain information of the response factor (PROF) but be clean of the independent factor (VALUE). We apply the same approach to MS and PM.

$$PROF = \gamma_0 + \gamma_1 VALUE + \mu$$
$$MS = \beta_0 + \beta_1 PM + \nu.$$

Table 4.6 Pearson and rank correlations of themes with forward returns for stocks in the Russell 1000, based on monthly data from January 31, 1995 to December 31, 2004

	Retf1	Retf3	Retf6	Retf9		Retf1	Retf3	Retf6	Retf9
Pearson cor.					**Rank cor.**				
VALUE	0.04	0.05	0.05	0.05	VALUE	0.03	0.05	0.06	0.07
PM	0.03	0.05	0.06	0.05	PM	0.03	0.04	0.05	0.04
PROF	0.04	0.05	0.06	0.06	PROF	0.04	0.06	0.07	0.08
EQ	0.01	0.02	0.03	0.03	EQ	0.01	0.02	0.03	0.03
MQ	0.03	0.05	0.06	0.06	MQ	0.02	0.04	0.06	0.06
MS	0.01	0.01	0.02	0.02	MS	0.01	0.02	0.02	0.03

Retf1 is one-month forward stock returns

Table 4.7 Pearson and rank correlations of themes for stocks in the Russell 1000, based on monthly data from January 31, 1995 to December 31, 2004

	VALUE	PM	PROF	EQ	MQ	MS
VALUE	1	−0.13	0.74	0.14	0.22	−0.12
PM	−0.14	1	0.01	0.07	0.04	0.48
PROF	0.68	0	1	0.27	0.3	0.04
EQ	0.11	0.07	0.25	1	0.2	0.11
MQ	0.22	0.04	0.31	0.2	1	−0.01
MS	−0.12	0.46	0.05	0.12	−0.01	1

The R scripts below show the relationship between the pairs and also a way to derive the residuals.

OLS, use residuals as a proxy

```
> summary(lm(PROF ~ VALUE, data = r1k.themes))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.001506   0.001304   1.155    0.248
VALUE       0.695197   0.001830 379.979   <2e-16 ***
---
Residual standard error: 0.4459 on 116890 degrees of freedom
Multiple R-squared:  0.5526,Adjusted R-squared:  0.5526
F-statistic: 1.444e+05 on 1 and 116890 DF,  p-value: < 2.2e-16

> summary(lm(MS ~ PM, data = r1k.themes))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0006538  0.0023841  -0.274    0.784
PM           0.5732920  0.0030951 185.227   <2e-16 ***
---
Residual standard error: 0.8151 on 116890 degrees of freedom
Multiple R-squared:  0.2269,Adjusted R-squared:  0.2269
F-statistic: 3.431e+04 on 1 and 116890 DF,  p-value: < 2.2e-16
```

We now use residuals from the OLS regressions and apply OLS to a multi-factor model

$$Retf9 = b_0 + b_1 VALUE + b_2 PM + b_3 PROF.resid + b_4 EQ + b_5 MQ + b_6 MS.resid + \epsilon.$$

OLS, multi-factor model with themes

```
> summary(lm(retf9 ~ VALUE + PM + PROF.resid + EQ
  + MQ +  MS.resid, data = r1k.themes))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.705e-02  1.239e-03  78.357  < 2e-16 ***
VALUE         2.971e-02  1.820e-03  16.327  < 2e-16 ***
PM            2.642e-02  1.657e-03  15.947  < 2e-16 ***
PROF.resid.   1.313e-02  2.963e-03   4.431 9.38e-06 ***
EQ            3.733e-03  2.147e-03   1.739   0.0821 .
MQ            2.853e-02  2.188e-03  13.038  < 2e-16 ***
MS.resid.     5.728e-05  1.543e-03   0.037   0.9704
---
Residual standard error: 0.4084 on 108723 degrees of freedom
  (8162 observations deleted due to missingness)
Multiple R-squared:  0.008207,Adjusted R-squared: 0.008152
F-statistic: 149.9 on 6 and 108723 DF,  p-value: < 2.2e-16
```

Using the t-values of the multi-factor OLS results, we employ the following formula as a reference to calculate the weights for themes:

$$W_k = \frac{\text{T-VALUE}_k}{\sum_{k=1}^{6} \text{T-VALUE}_k}.$$

OLS, t-value based theme weights

```
Build Alpha now:
Theme weights based on OLS t-values:
VALUE   PM     PROF.resid   EQ    MQ   MS.resid
0.3169  0.3095   0.08601   0.0337 0.2530  0.0007
```

With consideration of univariate and bivariate results, we build alpha scores as follows:

$$ALPHA = 0.25\,\text{VALUE} + 0.25\,\text{PM} + 0.10\,\text{PROF} + 0.10\,\text{EQ} + 0.20\,\text{MQ} + 0.10\,\text{MS}.$$

We then present simple summary statistics, correlations with forward returns, and an OLS regression model for ALPHA scores. The R scripts below show the results.

ALPHA

```
> summary(r1k.themes$ALPHA)
Statistics on the Efficacy of Alpha:
       V1
 Min.   :-2.227278
 1st Qu.:-0.221958
 Median : 0.006209
 Mean   : 0.002316
 3rd Qu.: 0.242810
 Max.   : 1.938843

> cor(r1k.themes$ALPHA, r1k.themes[, returns])
     retf1 retf3 retf6 retf9
[1,] 0.058  0.08 0.092  0.092

> cor(r1k.themes$ALPHA,r1k.themes[,returns],method="Spearman")
     retf1 retf3 retf6 retf9
[1,] 0.048 0.072 0.092 0.097

> summary(lm(retf9 ~ ALPHA, data = r1k.themes))
Residuals:
    Min      1Q  Median      3Q     Max
-1.1529 -0.2099 -0.0126  0.1751 27.6158

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.097047   0.001239   78.35   <2e-16 ***
ALPHA       0.090376   0.003042   29.71   <2e-16 ***
---
Residual standard error: 0.4084 on 108728 degrees of freedom
  (8162 observations deleted due to missingness)
Multiple R-squared:  0.008054,Adjusted R-squared: 0.008045
F-statistic: 882.8 on 1 and 108728 DF,  p-value: < 2.2e-16
```

As a brief summary, we present comments below about the results from R scripts for alpha diagnostics.

1. The summary statistics indicate that alpha values have mean/median zero and range from −2 to 2. Reflected in a stock selection portfolio, a stock with zero alpha indicates that the model does not have an opinion about its price movement (relative to the benchmark), and thus it should not be held in the portfolio from the pure alpha perspective. A stock with negative alpha should be avoided in a long-only portfolio or placed in short position for a long/short portfolio, Positive alphas should be associated with long-buy stocks.
2. The correlation values of ALPHA with forward returns are in the range of 5–10%, indicating high forecasting power of alpha values over the study period.
3. The OLS regression yields a t-value of 29.71, indicating a strong causal relationship between alpha values and nine-month forward returns. The R^2 is 0.8%, the highest we ever got comparing the regressions for signals and themes. The alpha can explain about 1% of all stock returns during this 10-year period.

So far, for a US large-cap-core stock selection strategy in Russell 1000, we have built signals, cleaned the data, applied proper treatments (such as removing industry biases), constructed themes, and built alpha values. We applied univariate, bivariate, and multi-factor OLS analysis during the process. We also made comments about the OLS properties (BLUE) and checked conditions for being BLUE using real-world data. However, we need to be cautious about several aspects of this alpha building process, including but not limited to:

- In-sample results. Note that all the analysis and results are based on an in-sample study. For a live portfolio, the future is unknown. An out-of-sample exploration is closer to a real-world investment.
- Industry-specific model. Each industry is different, but this does not necessarily mean we need to build a separate model for each industry. However, given the larger degree of difference between some industries, such as banking versus non-banking industries, an industry-specific banking model may be suitable to capture unique drivers and indicators for the stocks within the banking industry.
- Stationary versus dynamic. We apply the same procedure over the entire period and treat the signals, themes, and alpha all the same over time. However, we know that the relationships may change over time. Here we have ignored the dynamics.
- No risk involved in alpha construction. We either assume that risk information contained in signals is all the same or ignore the risk information during the alpha building process.
- Linear model. We adopt a simple linear approach for both themes and alpha construction. Of course, in the real world, the relationships between signals/themes/alpha are not linear. There are interactions among factors and nonlinear effects of factors on forward stock returns.

We will explore some of the issues above in detail in subsequent chapters with additional discussions, analysis, and possible solutions.

4.8 Commonly Used R Functions for Alpha Building

In this section, we introduce some simple R functions commonly used for alpha building in quantitative investing. We first introduce utility functions for data cleaning and signal treatment, then show multi-factor estimation for alpha construction.

4.8.1 R Functions: Data Cleaning and Signal Treatment

We discussed error and outliers in data in Chap. 2 and described the treatment procedure in the previous section. We now present simple functions for each step in the order of treatment.

$$\begin{aligned} Rawscores &\Rightarrow Missing \Rightarrow Truncation \Rightarrow Standardization \Rightarrow Winsorization \\ &\Rightarrow Industry\ demean \Rightarrow Re\text{-}standardization \Rightarrow Exclusion \\ &\Rightarrow Missing\ values \Rightarrow Distribution \Rightarrow Treatedscores \end{aligned}$$

The R functions for the above purposes are usually called utility functions as they can be used again and again in many cases of data cleaning and factor building. For a utility function, major inputs are typically data, ids and specific options for treatments. We will provide brief comments about them when appropriate.

Missing The presence of missing values can be serious if they exceed a certain percentage of the entire data. Unfortunately, this happens frequently. So, the first thing we need to do is to decide the maximum allowable percentage. For example, if the percentage of missing values is over 30%, we would simply drop the factor from the alpha building process.

Missing threshold

```
dataMiss <- function(data,varname,cut.point,theID="cusip")
{
   x=data
   cc=which(is.na(x[,varname]))
   if (sum(cc)>0)
   {
      ccc=length(cc)/dim(x)[1]
      x$missPercent=ccc
      if(ccc>cut.point)
      {
         x$missFlag=1  # date flag to avoid the factor
         x[,varname]=0  # assign zero for all names
```

```
      }
      else x$missFlag=0
      cat("Missing:", varname, ":", length(cc), "of",dim(x)[1],"\n")
      kkk=x[cc,c(theID,varname)]
      names(kkk)[2]="value"
      kkk$name=varname
      kkk$event="missing"
      print(kkk,row.names=F)
   }
   else
   {
     x$missPercent=0
     x$missFlag=0
     cat("Missing:", varname, ", no missing obs of ",dim(x)[1],"\n")
   }
   return(x)
}
```

Truncation, Standardization and Winsorization Once a signal passes the missing values threshold, we will clean up the data by removing errors and outliers and then standardize the scores and winsorize the outliers.

Regarding truncation, the criteria for removal can be based on either standard deviation or percentile. For example, we can set a rule that any stocks with values of more than 5 standard deviations (sigmas) will be removed or assigned NA (not available). Note that standard deviation itself is sensitive to outliers, we can overcome this issue by using a robust version of standard deviation calculation.

Truncation

```
dataTruncate <- function(data,varname,method="sigma",
                         LtruncPoint=5,RtruncPoint=5,theID="cusip")
{
  x=data

  ## deal with the left tail and right tail separately
  if(tolower(method)=="sigma")
  {
    tmp1<-mean(x[,varname],na.rm=T);
    tmp2<-sd(x[,varname],na.rm=T)
    minx=tmp1-LtruncPoint*tmp2
    maxx=tmp1+RtruncPoint*tmp2
    cc=which(x[,varname]>maxx | x[,varname]<minx)
    if (sum(cc)>0)
    {
      cat("Truncation: ",varname, "\n")
```

```
      kkk=x[cc,c(theID,varname)]
      names(kkk)[2]="value"
      kkk$name=varname
      kkk$event="truncation"
      print(kkk,row.names=F)
      x[cc,varname]<-NA
    }
    else cat("Truncation: ",varname," truncated ids = NA","\n")
  }
  else if(tolower(method)=="percentile")
  {
    minx<-as.vector(quantile(x[,varname],LtruncPoint/100,na.rm=T))
    maxx<-as.vector(quantile(x[,varname],(1-RtruncPoint/100),na.rm=T))
    cc=which(x[,varname]>maxx | x[,varname]<minx)
    if(sum(cc)>0)
    {
      cat("Truncation: ",varname, "\n")
      kkk=x[cc,c(theID,varname)]
      names(kkk)[2]="value"
      kkk$name=varname
      kkk$event="truncation"
      print(kkk,row.names=F)
      x[cc,varname]<-NA
    }
    else cat("Truncation: ",varname," truncated ids = NA","\n")
  }
  else  stop("Please select the correct truncation method! \n\n")
  return(x)
}
```

One option for standardization is to simply subtract the mean and divide by the standard deviation. This prepares for the winsorization step and eventually helps to compare factors apples-to-apples. Note that since standard deviation is sensitive to outliers, we add an option to use a robust version (R package *rrcov*) for the second moment calculation.

Standardization

```
dataStandardize <- function(data,varname,method="robust")
{
  # check the name in case of the repeated standar dization
  # robust version
  x = data
  ## If there are too few observations do not use robust
  if(method=="robust" & nrow(x) < 30) {
```

```
    method <- "simple"
  }

  if(method=="robust")
  {
   require(rrcov)
   #new.varname <- paste(varname,'.sdz',sep='');
   tmp <- CovSde(x[,varname]);
   x[,varname] <- x[,varname] - as.vector(tmp@center)
   x[,varname] <- x[,varname] / sqrt(as.vector(tmp@cov))
  }
  else
  {
     theMean=mean(x[,varname],na.rm=T)
     theStd=sd(x[,varname],na.rm=T)
     if(theStd != 0) x[,varname]=(x[,varname]-theMean)/theStd
  }
   return(x);
}
```

The winsorization is similar to truncation in the sense that both deal with outliers. The difference is that the former keeps outliers and shrinks them to a specified score, while the latter simply removes outliers.

Winsorization

```
dataWinsorize <- function(data,varname, Lwin=-3, Rwin=3,theID="cusip")
{
    x = data

    # deal with the left tail and right tail separately
    gt.index <- which(x[,varname] > Rwin);
    if(length(gt.index) > 0)
    {
       cat("Winsorization:",varname,"right tail:","\n")
       kkk=x[gt.index,c(theID,varname)]
       names(kkk)[2]="value"
       kkk$name=varname
       kkk$event="winsorization.right"
       print(kkk,row.names=F)
       x[gt.index,varname] <- Rwin
    }
    else cat("Winsorization: ",varname, "right tail = NA","\n")
```

```
    lt.index <- which(x[,varname] < Lwin);
    if(length(lt.index) > 0)
    {
       cat("Winsorization: ",varname,"left tail:","\n")
       kkk=x[lt.index,c(theID,varname)]
       names(kkk)[2]="value"
       kkk$name=varname
       kkk$event="winsorization.left"
       print(kkk,row.names=F)
       x[lt.index,varname] <- Lwin
    }
    else cat("Winsorization: ",varname, "left ids = NA","\n")

    return(x);
}
```

Industry Demean and Re-standardization This is a very important step. We explained the rationale in the previous section.

Industry demean

```
dataNeutralize <- function(data,varname, neutral.name)
{
   x=data
   # get the mean for each group
   cc=which(is.na(x[,neutral.name]))
   if(sum(cc)>0) x[cc,neutral.name] = "NA"
   aa=tapply(x[,varname],x[,neutral.name],mean,na.rm=T)
   theMean= data.frame(theMean = as.vector(aa), sss=names(aa))
   names(theMean)[2]=neutral.name
   x=merge(x,theMean,by=neutral.name,all.x=T)

   # get the demeaned score
   x[,varname]=x[,varname] - x$theMean
   mm=match("theMean",names(x))
   x=x[,-mm]
   return(x)
}
```

Re-standardization is employed to make sure factors have the same mean and standard deviation because industry demeaning may change the distribution of factor scores.

Exclusion Exclusion means to exclude a signal from a specified group. For example, cash flow factors do not make sense for banking, so we simply exclude cash-flow-based factors from themes and alpha building for banking companies.

Exclusion

```
dataExclude <-function(data, varname, col.exclude, exclude.name, theID="cusip")
{
  x=data
  # find the signal
  ss=match(varname,names(x))
  if(is.na(ss)) stop("varname does not exist in the data!\n")
  # exclude the group
  mm=match(col.exclude,names(x))
  if(is.na(mm)) stop ("col.exclude does not exist in the data!\n\n")
  nex=length(exclude.name)

  for(i in 1:nex)
  {
    cc=which(x[,mm]==exclude.name[i])
    if (sum(cc)>0)
    {
      x[cc,ss]=0
      cat("Exclusion: ",varname, "=", exclude.name[i], ": ",x[cc,theID],"\n")
    }
  }
  return(x)
 }
```

Missing Values and Distribution We deal with missing values again but now need to decide on how to treat the missing values: remove them or replace them with a preset score, such as zero, the mean, or the median? These preset values should be designed to associate stocks with missing values as neutral in a portfolio.

Missing replacement

```
dataReplaceMissing <- function(data,varname,theID="assetID",miss.fill="zero")
{
  x = data
  miss.index <- which(is.na(x[,varname]));

if(length(miss.index) > 0) {
 if(tolower(miss.fill)=="mean") x[miss.index,varname] <- mean(x[,varname],na.rm=T)
 if(tolower(miss.fill)=="median") x[miss.index,varname]<-median(x[,varname],na.rm=T)
 if(tolower(miss.fill)=="zero") x[miss.index,varname] <- 0
}
 return(x);
}
```

Distribution Matters This will depend on many considerations. For example, for an investment universe of small-cap companies or emerging markets, factor values are usually more wild than for large-cap companies in developed markets. The former may require ranking, while the latter can be normalized. We provide here a utility function with options for ranking, normalization, and maintaining the original distribution.

Distribution

```
dataDistribution <- function(data,varname, distrib)
{
  x=data
  if(distrib==1 |distrib==2)
  {
    #x$uniform=rank(x[,varname])/dim(x)[1]
 x$uniform=rank(x[,varname],na.last="keep",ties.method="average")/dim(x)[1]
    mm=match("uniform",names(x))
    names(x)[mm]=paste(varname,".unif",sep="")
  }
  if(distrib==2)
  {
    x$normal=pnorm(x[,mm],0,1)
    mm2=match("normal",names(x))
    names(x)[mm2]=paste(varname,".norm",sep="")
  }
  return(x)
}
```

Using the functions above, we clean the data and carry out treatment for the signal B/P for stocks in the Russell 1000 universe on December 31, 2004. We present the plot for the data at each stage in Fig. 4.10.

4.8.2 R Functions: Estimating a Multi-Factor Model with OLS

The previous section describes functions for data cleaning and factor treatment. In this section, we show how to estimate a multi-factor model with R scripts and functions. In particular, we show how to obtain OLS estimation results and output results using a function.

OLS and Its Attributes In R, the command to run OLS is *lm*. The protocol is

$$lm(y \sim x_1 + x_2, data = ABC),$$

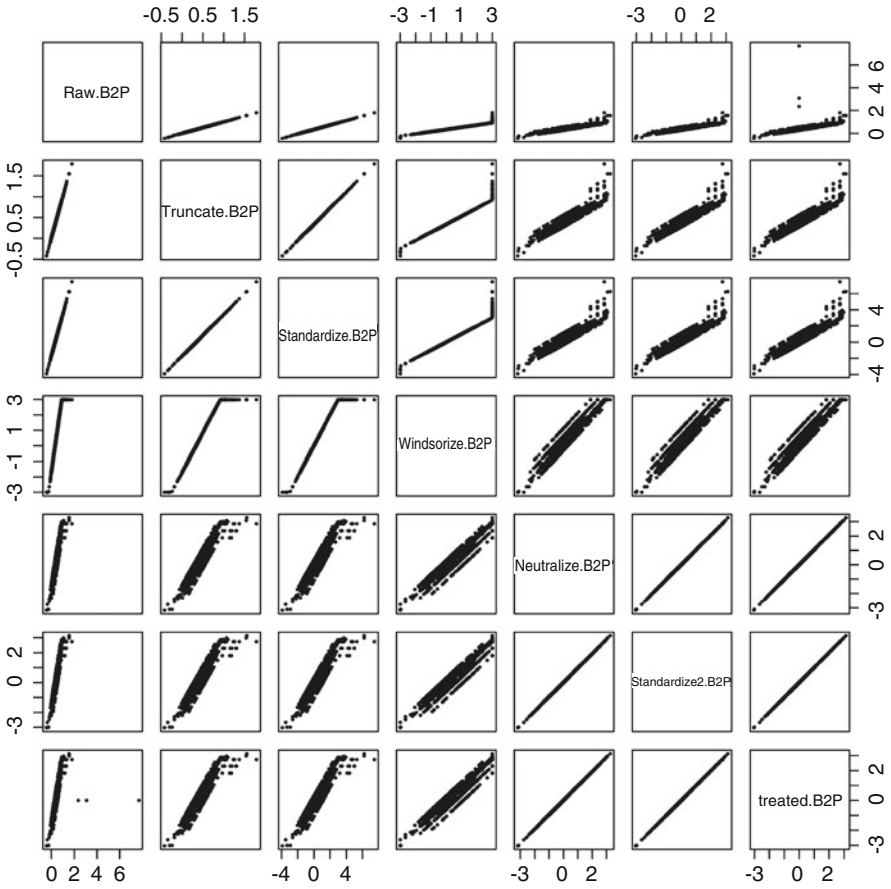

Fig. 4.10 The plot of B/P values at each step of the treatment for stocks in the Russell 1000 universe on December 31, 2004

where the data ABC contains columns of y, the response variable, and x_1 and x_2, the factors.

To get more information about the OLS results, we can use the R command *summary(ols.object)*. We use profitability signals as an example and show the R scripts below.

OLS example

```
> summary(lm(retf3 ~ FCF2EV + ROEFY0 + EBITDA2EV + roiCFO2CE,
           data = r1k.signals))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0327000  0.0006231  52.477  < 2e-16 ***
FCF2EV      0.0083439  0.0008187  10.192  < 2e-16 ***
ROEFY0      0.0036050  0.0007469   4.827 1.39e-06 ***
EBITDA2EV   0.0033314  0.0006872   4.848 1.25e-06 ***
roiCFO2CE   0.0022625  0.0008764   2.582  0.00984 **
---
Residual standard error: 0.2116 on 115355 degrees of freedom
  (1532 observations deleted due to missingness)
Multiple R-squared:  0.003079,Adjusted R-squared: 0.003045
F-statistic: 89.08 on 4 and 115355 DF,  p-value: < 2.2e-16
```

To obtain the components of the results above, we can use *coef* to get coefficients, *resid* to get residuals (the estimates for the errors), and *fitted* for fitted values $\hat{y}$. The R scripts below display the usage of these functions.

It is a good habit to start any R function with some explanatory comments. The "read.me" text for a function includes the purpose of the function and the prepared directories, folders and utility functions, etc.

Function, read.me part and preparation

```
################################################################################
##
## Function: Quant Investmenting, Chapter 4, build alpha
##
##   Input:  data.dir -- directory of the data
##          data.name -- the name of the input data with theme values
##             themes -- the name list of themes
##            returns -- the name list of returns
##           industry -- industry name, in case it is needed
##
##  Output: sink file -- univariate, bivariate and multi-factor OLS results
##          csv files -- correlation of themes, themes with returns
##                    -- correlations of alpha with returns
##                    -- final data frame with alpha values added
##
```

```
##    Note: Alpha values are built with the t-values of OLS model
##          residual values are used for PROF~VALUE and MS~PM
##
##   Codes: Lingjie Ma, 20181128
##
################################################################################

data.dir="/.../book/QuantInvesting/chapter3/factor.treatment/"
theme.list=c("VALUE","PM","PROF","EQ","MQ","MS")
return.list=c("retf1","retf3","retf6","retf9")

QI.chapter4.alpha <- function(data.dir,themes=them.list,
   returns=return.list, industry="sectorName")
{

}
```

We now present a sample function to illustrate the R commands for running OLS regression for a multi-factor model, obtaining components such as t-values and residuals, and calculating weights for factors based on OLS estimation results. Note that we also create a log file by command *sink* which is very helpful for debugging and recording purposes.

Sample R function: OLS for alpha model

```
QI.chapter4.alpha <- function(data.dir, data.name,
        themes=them.list, returns=return.list, industry="sectorName")
{
### get the data
r1k.themes=read.csv(paste(data.dir, data.name,sep=""), sep=",",header=T)

### start the log file
sink(paste(data.dir,"analysis/alpha.ols.txt",sep=""))

### Univariate results
cat("Univariate analysis, Summary of themes: \n")
print(summary(r1k.themes[,themes]))

### Bivariate, themes and with returns correlation
cat("Bivariate analysis, correlation of themes and with returns: \n")
## correlation with returns
 themes.corRetP=round(cor(r1k.themes[,themes],r1k.
               themes[,returns],use="complete.obs"),2)
 themes.corRetS=round(cor(r1k.themes[,themes],r1k.
                  themes[,returns],use="complete.obs",method="spearman"),2)

 write.table(themes.corRetP, paste(data.dir,"analysis/
               r1k.themes.corRetP.csv",sep=""),sep=",")
```

```
 write.table(themes.corRetS, paste(data.dir,"analysis/
              r1k.themes.corRetS.csv",sep=""),sep=",")

 ## correlation between signals
 themes.corP=round(cor(r1k.themes[,themes], use="complete.obs"),2)
 themes.corS=round(cor(r1k.themes[,themes],
      use="complete.obs",method="spearman"),2)

 write.table(themes.corP, paste(data.dir,"analysis/r1k.
                      themes.corP.csv",sep=""),sep=",")
 write.table(themes.corS, paste(data.dir,"analysis/r1k.
                      themes.corS.csv",sep=""),sep=",")

### Multi-factor model, OLS results: get resid
cat("Multi-factor, OLS of returns on themes: \n")
prof.ols=summary(lm(PROF~VALUE,data=r1k.themes))
print(prof.ols)
r1k.themes$PROFexVALUE.resid=prof.ols$resid

ms.ols=summary(lm(MS~PM,data=r1k.themes))
print(ms.ols)
r1k.themes$MSexPM.resid=ms.ols$resid

themes.ols=summary(lm(retf9~ VALUE+PM + PROF.resid
         + EQ+MQ+MS.resid,data=r1k.themes))
print(themes.ols)

### OLS, obtain t-values, calculate weight and build alpha scores
cat("Build Alpha now: \n")
theme.Tvalue=themes.ols$coef[-1,3]
theme.Tweight=theme.Tvalue/sum(abs(theme.Tvalue))
cat("Theme weights based on OLS t-values:\n")
print(theme.Tweight)

### Statistics on the Efficacy of Alpha
cat("Statistics on the Efficacy of Alpha: \n")
r1k.themes$ALPHA =  as.matrix(r1k.themes[,themes],
  ncol=6) %*% matrix(as.vector(theme.Tweight),ncol=1)
print(summary(r1k.themes$ALPHA))
print(round(cor(r1k.themes$ALPHA,r1k.themes[,returns],
                                  use="complete.obs"),3))
print(round(cor(r1k.themesALPHA,r1k.themes[,returns],
              use="complete.obs",method="spearman"),3))
print(summary(lm(retf9~ALPHA,data=r1k.themes)))

### Output the data with alpha values
write.table(r1k.themes, paste(data.dir,"treateddata/r1k.factors.
                alpha.csv",sep=""),sep=",",row.names=F)

sink()
}
```